

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-244557

(43)Date of publication of application : 30.08.2002

(51)Int.Cl.

G09C 1/00

H04L 9/32

(21)Application number : 2001-039572

(71)Applicant : ATR ADAPTIVE COMMUNICATIONS RES LAB

(22)Date of filing : 16.02.2001

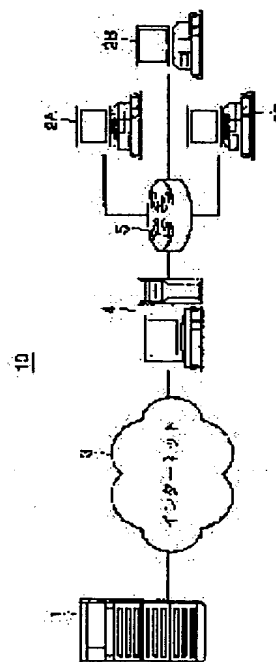
(72)Inventor : KIRIMOTO NAOKI  
YAMAZAKI TATSUYA

## (54) CRYPTOGRAPHIC COMMUNICATION SYSTEM AND AUTHENTICATION METHOD USED THEREFOR

## (57)Abstract:

**PROBLEM TO BE SOLVED:** To provide a cryptographic communication system permitting mutual authentication without transmitting own certificates to a server from clients, and to provide an authentication method used therefor.

**SOLUTION:** The cryptographic communication system 10 comprises a server 1, client servers 2A, 2B, 2C, the Internet network 3, CA Proxy(Certificate Authority Proxy) 4, and a coupler 5. The CA Proxy 4 is arranged between the Internet network 4 and the coupler 5. In response to a request for submittal of the digital certificates of the client servers 2A, 2B, 2C from the server 1, the CA Proxy 4 transmits to the server 1 their own digital certificates presenting that the client servers 2A, 2B, 2C are authorized servers, irrespective of whether or not the client servers 2A, 2B, 2C possess the digital certificates.



## LEGAL STATUS

[Date of request for examination] 09.03.2001

[Date of sending the examiner's decision of rejection] 24.08.2004

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2002-244557

(P2002-244557A)

(43) 公開日 平成14年8月30日 (2002.8.30)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード <sup>*</sup> (参考)
G 0 9 C 1/00	6 4 0	G 0 9 C 1/00	6 4 0 Z 5 J 1 0 4
H 0 4 L 9/32		H 0 4 L 9/00	6 7 5 D

審査請求 有 請求項の数 9 O L (全 18 頁)

(21) 出願番号 特願2001-39572 (P2001-39572)

(22) 出願日 平成13年2月16日 (2001.2.16)

(71) 出願人 396011680

株式会社エイ・ティ・アール環境適応通信  
研究所

京都府相楽郡精華町光台二丁目2番地2

(72) 発明者 桐本 直樹

京都府相楽郡精華町光台二丁目2番地2  
株式会社エイ・ティ・アール環境適応通信  
研究所内

(74) 代理人 100064746

弁理士 深見 久郎 (外4名)

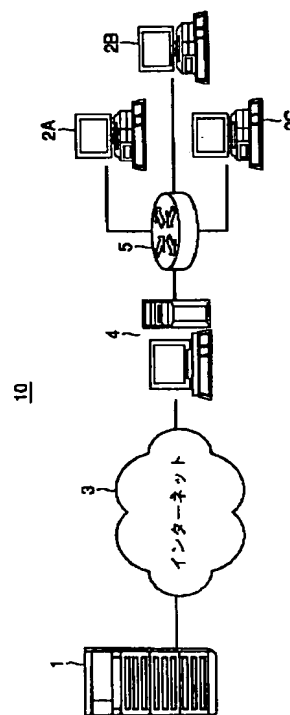
最終頁に続く

(54) 【発明の名称】 暗号通信システムおよびそれに用いる認証方法

(57) 【要約】

【課題】 クライアントがサーバに対して自己の証明書を送信しなくても、相互認証が可能な暗号通信システムおよびそれに用いる認証方法を提供する。

【解決手段】 暗号通信システム10は、サーバ1と、クライアントサーバ2A、2B、2Cと、インターネット網3と、CA Proxy (Certificate Authority Proxy) 4と、結合器5とを備える。CA Proxy 4は、インターネット網3と結合器5との間に配置される。CA Proxy 4は、サーバ1からクライアントサーバ2A、2B、2Cの電子証明書の提出要求に応じて、クライアントサーバ2A、2B、2Cが電子証明書を保持しているか否かに拘わらず、サーバ1に対してクライアントサーバ2A、2B、2Cが正規のサーバであることを示す自己の電子証明書をサーバ1へ送信する。



## 【特許請求の範囲】

【請求項 1】 複数の層構造から成るプロトコルを用いて、所定の暗号方式によって暗号化された暗号データの通信を行なう暗号通信システムであって、データまたは前記暗号データを送受信する第 1 のサーバと、

前記第 1 のサーバとの間で前記データまたは暗号データを送受信する第 2 のサーバと、

前記第 1 のサーバと前記第 2 のサーバとの通信を中継する第 3 のサーバとを備え、

前記第 2 のサーバの認証時、前記第 3 のサーバは、前記第 1 のサーバからの前記第 2 のサーバの証明書の要求に応じて、前記第 2 のサーバが正規のサーバであることを示す代理証明書を前記第 2 のサーバに代わって前記第 1 のサーバへ送信する、暗号通信システム。

【請求項 2】 前記第 3 のサーバは、自己の証明書を前記代理証明書として前記第 1 のサーバへ送信する、請求項 1 に記載の暗号通信システム。

【請求項 3】 前記第 3 のサーバは、前記第 2 のサーバの証明書の要求を前記第 2 のサーバへ送信し、前記第 2 のサーバから前記第 2 のサーバの証明書を受信すると前記代理証明書を前記第 1 のサーバへ送信する、請求項 1 または請求項 2 に記載の暗号通信システム。

【請求項 4】 前記第 3 のサーバは、前記第 2 のサーバの証明書の要求を前記第 2 のサーバへ送信し、前記第 2 のサーバが証明書を保持しないことを示す情報を前記第 2 のサーバから受信すると前記代理証明書を前記第 1 のサーバへ送信する、請求項 1 または請求項 2 に記載の暗号通信システム。

【請求項 5】 前記第 3 のサーバは、前記第 1 のサーバから受信した前記第 1 のサーバの証明書を証明書廃棄リストと照合し、前記第 1 のサーバの証明書が廃棄されているとき前記第 1 のサーバの証明書が無効であることを示す情報を前記第 1 のサーバへ送信する、請求項 1 から請求項 4 のいずれか 1 項に記載の暗号通信システム。

【請求項 6】 前記第 3 のサーバは、前記第 1 のサーバから受信した前記第 1 のサーバの証明書を証明書廃棄リストと照合し、前記第 1 のサーバの証明書が廃棄されていないことを確認すると前記代理証明書を前記第 1 のサーバへ送信する、請求項 1 から請求項 4 のいずれか 1 項に記載の暗号通信システム。

【請求項 7】 前記第 3 のサーバは、前記第 1 のサーバと前記第 2 のサーバとの間の通信を制御する通信制御部と、前記通信制御部を介して受取った前記第 1 のサーバの証明書を前記証明書廃棄リストと照合する証明書照合部と、前記証明書照合部からの照合結果に基づいて、前記第 1 のサーバを認証し、または前記第 1 のサーバの証明書を無効と判定する認証部と、

前記通信制御部を介して前記第 2 のサーバの証明書または前記第 2 のサーバが証明書を保持しない情報を受取ると、前記代理証明書を前記第 1 のサーバへ送信する代理応答部を含む、請求項 5 または請求項 6 に記載の暗号通信システム。

【請求項 8】 第 1 のサーバと第 2 のサーバとの間における認証方法であって、

前記第 2 のサーバの証明書の送信要求を前記第 1 のサーバから受信する第 1 のステップと、

10 前記第 2 のサーバの証明書に代えて代理証明書を前記第 1 のサーバへ送信する第 2 のステップとを備える認証方法。

【請求項 9】 前記第 2 のステップにおいて、前記代理証明書は、前記第 2 のサーバが証明書を保持するか否かに拘わらず前記第 1 のサーバへ送信される、請求項 8 に記載の認証方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】この発明は、暗号通信システムに関し、特に、サーバからのクライアントの証明書の要求に対して代理応答する暗号通信システムおよびそれに用いる認証方法に関するものである。

## 【0002】

【従来の技術】現在、インターネットを用いて各種の情報が通信されている。このインターネットを用いた通信においては、図 10 に示すように、2 つのコンピュータ 100、110 は、OSI (Open System Interconnection) 参照モデル 130 を用いて各種の情報を送受信する。コンピュータ 100 は、ケーブル、無線等の物理的接続 120 によってコンピュータ 110 とデータ等の各種の情報をやり取りする。OSI 参照モデル 130 は、物理層、データリンク層、ネットワーク層、トランスポート層、セッション層、プレゼンテーション層、およびアプリケーション層の 7 層から成る。物理層が最下層であり、アプリケーション層が最上層である。また、第 1 層から第 4 層までが下位層であり、第 5 層から第 7 層が上位層である。コンピュータ 100 がコンピュータ 110 と通信を行なうとき、各層は、それぞれ、機能の異なるプロトコルであるが、互いに同じ層のやり取りを意識することによって結果的に全体の通信プロトコルが成立する。

【0003】図 11 は、OSI 参照モデル 130 の各層の機能を示したものである。最上層のアプリケーション層は、コンピュータ 100、110 を操作するユーザが使用するアプリケーション (サービス) と下の層との橋渡しを行なう。例えば、コンピュータ 100、110 のユーザが通信すべきメッセージとして「こんにちは」とキーボードから入力したとき、これを下の層に引渡したり、逆に下の層からメッセージが届いた場合は、これをアプリケーションとして認識する働きをする。データ通

## 3

信用のアプリケーションには、ファイル転送や電子メールの送受信等、様々なものであるが、アプリケーション層では、届いたデータがこれらのアプリケーションのうち、どのアプリケーションのものかを判断し、受信側で該当する正しいアプリケーションに引渡す。また、送信される情報では、送信先のどのアプリケーションへ渡されるものかが明確になっている。従って、アプリケーション層は、これらの交通整理を的確に行ない、情報自体を認識し、制御する。

【0004】プレゼンテーション層は、上の層から渡された情報を通信に適した形に変換して下の層に引渡したり、下の層から渡された情報をアプリケーション層に適した形に変換して上の層に引渡す。例えば、プレゼンテーション層は、上述した「こんにちは」というメッセージを符号して下の層に引渡し、下の層から渡された符号化されたデータを「こんにちは」というメッセージに変換して上の層へ渡す。セッション層は、プレゼンテーション層で符号化されたデータを相手に送信する機能を果たす。つまり、セッション層は、データを相手に送信する際、相手との通信経路を確立したり、通信方法を決定する。

【0005】トランスポート層は、通信情報の質を高めるための通信制御を行なう。具体的には、トランスポート層は、相手との通信においてデータが欠落したとき、その欠落したデータを再送信してもらうように依頼する。ネットワーク層は、データを送信する際の送信元と送信先とを決定する。つまり、ネットワーク層は、データにヘッダを付加し、そのヘッダに送信元のアドレスと送信先のアドレスとを書込む。データリンク層は、ネットワーク層で設定された相手のアドレスに基づいて、データを次に送信すべき宛先を管理する。物理層は、データリンク層から渡されたビット情報を実際に伝送するための電気信号に変換したり、受信した電気信号をビット情報に変換する。

【0006】上述した7層から成るOSI参照モデル130を用いて2つのコンピュータ100、110間でデータが通信される。このようなインターネットを用いたデータ通信においては、データのセキュリティが重要な問題であり、データを暗号化して通信することが行なわれている。インターネットを用いた暗号通信で一般的に使用されているのは、SSL(Secure Socket Layer)暗号通信である。SSL暗号通信は、公開鍵を基盤としたセキュリティ・インフラストラクチャーであるPKI(Public Key Infrastructure)の実装暗号化通信方式である。そして、SSL暗号通信においては、送信データの暗号化に共通鍵暗号方式が用いられ、クライアントとサーバとが送信データの暗号化に用いる共通鍵を共有するために公開鍵暗号方式が用いられている。

【0007】インターネットによる通信において、不正

## 4

なアクセスなどによる重要データの漏洩・破壊を防ぐためにファイアウォールを設け、通過するデータを制御することが行なわれている。しかし、SSL暗号通信の場合、OSI参照モデルの上層部が暗号化されているため、暗号化されていない平文での通信に比べファイアウォールで十分な通信制御を行なうことができない。このような問題を解決するために、“情報処理学会第58回(平成11年前期)全国大会5N-6, 3-333~3-334”には、インターネットとの暗号データとのやり取りをサーバに代わって行なうSSL代理応答システムが開示されている。図12は、このSSL代理応答システムを示したものである。SSL代理応答システム200は、インターネット210と、ファイアウォール220と、代理サーバ230と、サーバ240とから成る。ファイアウォール220は、インターネット210とサーバ240との間に配置され、代理サーバ230はファイアウォール220に接続されている。インターネット210からサーバ240へ暗号データが送信される時、ファイアウォール220は暗号データを代理サーバ230へ送信する。そして、代理サーバ230は、受信した暗号データを復号し、その復号した平文のデータをファイアウォール220へ送信する。ファイアウォール220は、代理サーバ230からの平文のデータに対して所定の制御を行なった後、サーバ240へ送信する。

【0008】サーバ240からデータを送信するとき、サーバ240は、平文のデータをファイアウォール220へ送信する。ファイアウォール220は、受信したデータに対して所定の制御を行なった後、データを代理サーバ230へ送信する。代理サーバ230は、受信したデータを暗号化して暗号データをファイアウォール220へ送信する。ファイアウォール220は、受信した暗号データをインターネット210を介して送信する。

【0009】このSSL代理応答システム200においては、代理サーバ230がサーバ240に代わってデータの暗号化および復号化を行なうことにより、ファイアウォール220によるアプリケーション層の通信制御が可能にしている。また、暗号データの通信においては、通信相手が正規の相手であることを相互に認証することがセキュリティの面から重要であるが、SSL代理応答システム200においては、代理サーバ230は、証明書および秘密鍵をサーバ240から譲り受け、サーバ240に代わって暗号認証通信を行なう。

【0010】また、暗号データの通信において相互認証するとき、その相互認証に用いる証明書が漏洩したとき、つまり、サーバの公開鍵やユーザの個人情報などを暗号化する秘密鍵(この秘密鍵は認証機関によって認証されている)が漏洩した場合、不正に侵入してきた相手と暗号データを通信することになり、重要なデータが外部に漏洩する。このような問題を解決するために、“コンピ

ユータセキュリティ 7-4 (2000. 1. 21) p 19~24”には、認証辞書 (Authenticated Dictionary) を用いて通信相手から送信されてきた証明書が廃棄された証明書でないことを確認するシステムが開示されている。この認証辞書は、証明書が漏洩したことを示す証明書廃棄リスト (CRL: Certificate Revocation List) と等価な証明書の廃棄情報を持つものである。このシステムにおいては、SSL暗号通信における通信路の確立時にサーバから送信された証明書の廃棄情報をクライアントが確認する。そして、クライアントは、受信した証明書が廃棄情報に含まれていなければ、その証明書に基づいて通信相手を認証する。

#### 【0011】

【発明が解決しようとする課題】しかし、上述した従来の暗号データの通信方式においては、通信相手の認証時、暗号データの通信を行なう送信者と受信者との間で、自己の証明書を提示し合って認証するものである。そして、証明書には、ユーザの個人情報も含まれるため、通信相手の認証時に個人情報が相手に知られるという問題がある。図12に示すSSL代理応答システム200においても、代理サーバ230が暗号認証通信を行なうが、その際に送信するのはサーバ240の証明書である。したがって、サーバ240の所有者の個人情報が通信相手に知られる。

【0012】また、従来の暗号データの通信方式においては、通信相手の相互認証は証明書に基づいており、いずれか一方の秘密鍵が漏洩した場合、第三者のなりすましが危惧される。しかし、現在のSSL暗号通信の確立段階で、証明書の有効性を検証する証明書廃棄リスト (CRL) との照合が行なわれていないため、第三者のなりすましによる不正な暗号通信を防止できないという問題がある。

【0013】そこで、本発明は、かかる問題を解決するためになされたものであり、その目的は、クライアントがサーバに対して自己の証明書を送信しなくても、相互認証が可能な暗号通信システムおよびそれに用いる認証方法を提供することである。

【0014】また、本発明の別の目的は、さらに、漏洩した証明書を送信した相手との通信を防止できる暗号通信システムを提供することである。

#### 【0015】

【課題を解決するための手段】この発明による暗号通信システムは、複数の層構造から成るプロトコルを用いて、所定の暗号方式によって暗号化された暗号データの通信を行なう暗号通信システムであって、データまたは暗号データを送受信する第1のサーバと、第1のサーバとの間でデータまたは暗号データを送受信する第2のサーバと、第1のサーバと第2のサーバとの通信を中継する第3のサーバとを備え、第2のサーバの認証時、第3

のサーバは、第1のサーバからの第2のサーバの証明書の要求に応じて、第2のサーバが正規のサーバであることを示す代理証明書を第2のサーバに代わって第1のサーバへ送信する。

【0016】この発明による暗号通信システムにおいては、第1のサーバが第2のサーバに対して証明書の送信を要求したとき、第3のサーバは、第2のサーバに代わって代理証明書を第1のサーバへ送信する。

【0017】したがって、この発明によれば、第2のサーバは、個人情報を第1のサーバへ送信しなくても第1のサーバとの間で暗号通信を行なうことができる。

【0018】好ましくは、暗号通信システムにおいて、第3のサーバは、自己の証明書を代理証明書として第1のサーバへ送信する。

【0019】第3のサーバは、第2のサーバの証明書の代わりに自己の証明書を第1のサーバへ送信する。そして、第1のサーバは、代理証明書に基づいて第2のサーバを正規のサーバとして認証する。

【0020】したがって、この発明によれば、第2のサーバは、自己の証明書を第1のサーバへ送信しなくても正規のサーバとして第1のサーバとの間で暗号通信を行なうことができる。

【0021】好ましくは、暗号通信システムにおいて、第3のサーバは、第1のサーバから受信した第2のサーバの証明書の要求を第2のサーバへ送信し、第2のサーバから第2のサーバの証明書を受信すると代理証明書を第1のサーバへ送信する。

【0022】第2のサーバは、第1のサーバからの第2のサーバの証明書の送信要求を受取り、自己の証明書を第3のサーバへ送信する。そうすると、第3のサーバは、第2のサーバから受信した証明書に代えて代理証明書を第1のサーバへ送信する。

【0023】したがって、この発明によれば、第2のサーバが証明書を保持している場合でも第2のサーバの個人情報を第1のサーバに公表せずに第1のサーバと第2のサーバとの間で暗号通信を行なうことができる。

【0024】好ましくは、暗号通信システムにおいて、第3のサーバは、第2のサーバの証明書の要求を第2のサーバへ送信し、第2のサーバが証明書を保持しないことを示す情報を第2のサーバから受信すると代理証明書を第1のサーバへ送信する。

【0025】第2のサーバは、第1のサーバからの第2のサーバの証明書の送信要求を受取り、証明書を保持していないことを第3のサーバへ送信する。そうすると、第3のサーバは、自己が保持する代理証明書を第1のサーバへ送信する。

【0026】したがって、この発明によれば、第2のサーバが証明書を保持していない場合でも、正規のサーバとして第1のサーバとの間で暗号通信を行なうことができる。

【0027】好ましくは、暗号通信システムにおいて、第3のサーバは、第1のサーバから受信した第1のサーバの証明書を証明書廃棄リストと照合し、第1のサーバの証明書が廃棄されているとき第1のサーバの証明書が無効であることを示す情報を第1のサーバへ送信する。

【0028】第1のサーバの証明書が証明書廃棄リストに含まれるとき、第1のサーバへ証明書の無効通知が送信される。

【0029】したがって、この発明によれば、不正な相手との暗号通信を防止できる。好ましくは、暗号通信システムにおいて、第3のサーバは、第1のサーバから受信した第1のサーバの証明書を証明書廃棄リストと照合し、第1のサーバの証明書が廃棄されていないことを確認すると代理証明書を第1のサーバへ送信する。

【0030】第2のサーバの証明書の送信を要求した第1のサーバが正規であることが確認されると、第3のサーバは、代理証明書を第1のサーバへ送信する。

【0031】したがって、この発明によれば、暗号通信におけるセキュリティを向上させることができる。

【0032】好ましくは、第3のサーバは、第1のサーバと第2のサーバとの間の通信を制御する通信制御部と、通信制御部を介して受取った第1のサーバの証明書を証明書廃棄リストと照合する証明書照合部と、証明書照合部からの照合結果に基づいて、第1のサーバを認証し、または第1のサーバの証明書を無効と判定する認証部と、通信制御部を介して第2のサーバの証明書または第2のサーバが証明書を保持しない情報を受取ると、代理証明書を第1のサーバへ送信する代理応答部とを含む。

【0033】第3のサーバは、第1のサーバと第2のサーバとの間の通信を制御するとともに、証明書の証明書廃棄リストとの照合、証明書の認証、および代理応答を行なう。

【0034】したがって、この発明によれば、暗号通信を行なう2つのサーバの間に、代理応答等を行なうサーバを設けることによって個人情報を隠匿して暗号通信を行なうことができる。

【0035】また、この発明による認証方法は、第1のサーバと第2のサーバとの間における認証方法であって、第2のサーバの証明書の送信要求を第1のサーバから受信する第1のステップと、第2のサーバの証明書に代えて代理証明書を第1のサーバへ送信する第2のステップとを備える。

【0036】この発明による認証方法においては、第1のサーバからの証明書の送信要求に対して代理証明書が第1のサーバへ送信されて第2のサーバの認証が行なわれる。

【0037】したがって、この発明によれば、個人情報を相手に公開せずとも相互認証を行なうことができる。

【0038】好ましくは、第2のステップにおいて、代

理証明書は、第2のサーバが証明書を保持するか否かに拘わらず第1のサーバへ送信される。

【0039】第2のサーバが認証機関によって認証された証明書を保持しているか否かに関係なく代理証明書が第1のサーバへ送信され、第2のサーバの認証が行なわれる。

【0040】したがって、この発明によれば、証明書を保持しないサーバでも正規のサーバとして暗号通信を行なうことができる。

10 【0041】

【発明の実施の形態】以下、本発明の実施の形態について図面を参照しながら詳細に説明する。なお、図中同一または相当部分には同一符号を付してその説明は繰返さない。

【0042】図1は、本発明による暗号通信システムの概略ブロック図である。暗号通信システム10は、サーバ1と、クライアントサーバ2A、2B、2Cと、インターネット網3と、CA Proxy (Certificate Authority Proxy) 4と、結合器5とを備える。サーバ1は、インターネット網3に接続される。クライアントサーバ2A、2B、2Cは、結合器5に接続される。CA Proxy 4は、インターネット網3と結合器5との間に配置される。

【0043】サーバ1は、後述する方法によってインターネット網3、CA Proxy 4、および結合器5を介してクライアントサーバ2A、2B、2Cへデータまたは暗号データを送信し、またはクライアントサーバ2A、2B、2Cからデータまたは暗号データを受信する。インターネット網3は、サーバ1からのデータまたは暗号データをCA Proxy 4へ送信し、CA Proxy 4からのデータまたは暗号データをサーバ1へ送信する。

【0044】CA Proxy 4は、インターネット網3からのデータまたは暗号データを結合器5を介してクライアントサーバ2A、2B、2Cへ送信する。また、CA Proxy 4は、クライアントサーバ2A、2B、2Cからのデータまたは暗号データを結合器5を介して受信し、その受信したデータまたは暗号データをインターネット網3を介してサーバ1へ送信する。さらに、CA Proxy 4は、サーバ1から受信した電子証明書を証明書廃棄リスト(CRL)と照合し、受信した電子証明書がCRLに含まれていないとき、受信した電子証明書に基づいてサーバ1を正規の通信相手として認証する。また、さらに、CA Proxy 4は、サーバ1からクライアントサーバ2A、2B、2Cの電子証明書の提出要求に応じて、自己の電子証明書をサーバ1へ送信する。即ち、CA Proxy 4は、クライアントサーバ2A、2B、2Cが電子証明書を保持しているか否かに拘わらず、サーバ1に対してクライアントサーバ2A、2B、2Cが正規のサーバであることを示す自己の

電子証明書を送信する。つまり、CA Proxy 4は、クライアントサーバ2A、2B、2Cに代わってサーバ1に対して電子証明書の代理応答を行なう。これにより、クライアントサーバ2A、2B、2Cは、自己の電子証明書をサーバ1へ送信しなくても、相互認証ができ、サーバ1との間で通信路を確立できる。

【0045】結合器5は、CA Proxy 4からのデータまたは暗号データをクライアントサーバ2A、2B、2Cの各々へ振り分ける。

【0046】図2は、サーバ1、クライアントサーバ2A、2B、2C、およびCA Proxy 4の機能ブロックを示したものである。サーバ1は、Record Protocol部11と、Handshake Protocol部12と、Change Cipher Spec Protocol部13と、Alert Protocol部14と、Application Data Protocol部15と、アプリケーション部16とを含む。Record Protocol部11、Handshake Protocol部12、Change Cipher Spec Protocol部13、Alert Protocol部14、およびApplication Protocol部15はSSLプロトコルを構成する。SSLプロトコルは、図10に示したセッション層のプロトコルであり、アプリケーション部16は、図10に示したセッション層よりも上位のプロトコルである。

【0047】Record Protocol部11は、アプリケーション部16からのデータを圧縮／暗号化して、クライアントサーバ2A、2B、2Cへ送信する。また、クライアント2A、2B、2Cから受信した暗号データを復号化／伸張してアプリケーション部16へ引渡す。Handshake Protocol部12は、暗号化アルゴリズム、暗号鍵、電子証明書など、暗号データの通信を開始するために必要なパラメータを通信相手と交渉し、決定する。

【0048】Change Cipher Spec Protocol部13は、Handshake Protocol部12で決定された、新しい暗号化通信パラメータの利用開始を通信相手に通知し、自らも利用を開始する。Alert Protocol部14は、通信中に発生したイベントやエラーを通信相手に通知する。Application Data Protocol部15は、現在、有効な暗号化通信パラメータに従って、アプリケーションデータを透過的に送受信する。アプリケーション部16は、SSLプロトコルのApplication Data Protocol部15からのデータを受取って、その受取ったデータを処理する。また、アプリケーション部16は、新たに作成したデータをSSLプロトコルのApplication Data Protocol部15へ引渡す。

【0049】クライアントサーバ2A、2B、2Cは、Record Protocol部21と、Handshake Protocol部22と、Change Cipher Spec Protocol部23と、Alert Protocol部24と、Application Data Protocol部25と、アプリケーション部26とを含む。Record Protocol部21、Handshake Protocol部22、Change Cipher Spec Protocol部23、Alert Protocol部24、およびApplication Data Protocol部25は、SSLプロトコルを構成する。Record Protocol部21、Handshake Protocol部22、Change Cipher Spec Protocol部23、Alert Protocol部24、およびApplication Data Protocol部25は、それぞれ、サーバ1のRecord Protocol部11、Handshake Protocol部12、Change Cipher Spec Protocol部13、Alert Protocol部14、およびApplication Data Protocol部15と同じ機能を果たす。また、アプリケーション部26は、サーバ1のアプリケーション部16に対応する。

【0050】CA Proxy 4は、通信プロトコルキャプチャ部41と、CRLチェック部42と、認証部43と、代理応答部44とを含む。通信プロトコルキャプチャ部41は、サーバ1とクライアントサーバ2A、2B、2Cとの間で行なわれる通信を監視するとともに、CRLチェック部42、認証部43、および代理応答部44で処理すべき通信プロトコルをそれぞれに振り分ける。CRLチェック部42は、第三者である認証機関（図示せず）が作成した証明書廃棄リスト（CRL）を認証機関から取得して保持しており、通信プロトコルキャプチャ部41がサーバ1から受信したサーバ1の電子証明書を受け、サーバ1の電子証明書を証明書廃棄リストと照合し、その照合結果を通信プロトコルキャプチャ部41へ出力する。なお、CRLチェック部42は、認証機関から証明書廃棄リストを定期的に取得し、証明書廃棄リストの更新を行なう。

【0051】認証部43は、CA Proxy 4の電子証明書を保持しており、サーバ1からクライアントサーバ2A、2B、2Cの電子証明書の提出を要求されると、クライアントサーバ2A、2B、2CはCA Proxy 4の支配下にあるクライアントサーバであることの認証を行なう。この場合、認証部43は、クライアントサーバ2A、2B、2Cの電子証明書に代えてCA Proxy 4の電子証明書を代理応答部44へ出力する。代理応答部44は、クライアントサーバ2A、2

B, 2Cから”Client Certificate”メッセージまたは”No Certificate”メッセージを通信プロトコルキャプチャ部41を介して受信すると、認証部43から受取ったCA Proxy 4の電子証明書をクライアントサーバ2A, 2B, 2Cに代わってサーバ1へ送信する。

【0052】なお、サーバ1およびクライアントサーバ2A, 2B, 2CにおけるSSLプロトコルはIETF (International Engineering Task Force) によって公開されている機能と同じである。

【0053】図3は、電子証明書の構成を示す概略ブロック図である。電子証明書50は、バージョン51と、シリアル番号52と、証明書発行者53と、発行者ユニーク識別子54と、証明対象ユーザ55と、ユーザユニーク識別子56と、証明対象公開鍵アルゴリズム57と、証明対象公開鍵58と、証明書有効期限59と、証明書拡張60と、署名アルゴリズム61と、署名62とを含む。バージョン51は、証明書50の構成要素を規定するものである。シリアル番号52は、その証明書が何番目に発行されたかを示す番号である。発行者ユニーク識別子54は、証明書を発行する認証機関を識別するための情報であり、認証機関に固有のIDが書込まれる。証明対象ユーザ55は、認証機関に自己の公開鍵の認証を依頼するユーザの名前である。ユーザユニーク識別子56は、認証機関に自己の公開鍵の認証を依頼するユーザを識別するための情報であり、ユーザIDが書込まれる。証明対象公開鍵アルゴリズム57は、ユーザが認証を依頼した公開鍵を生成するためのアルゴリズムである。証明対象公開鍵58は、ユーザに依頼されて、認証機関が認証する公開鍵である。証明書有効期限59は、電子証明書50が有効である期間である。証明書拡張60は、将来、各種の情報を電子証明書に格納するための枠組みを与えるものである。署名アルゴリズム61は、認証機関が署名するとき、つまり、電子証明書50のバージョン51から署名アルゴリズム61の各情報を認証機関の秘密鍵で暗号化し、または認証機関の復号鍵で復号するときのアルゴリズムである。署名62は、認証機関が電子証明書50のバージョン51から署名アルゴリズム61の各情報を自己の秘密鍵で暗号化していることを示す情報である。暗号通信システム10は、データを暗号化する共通鍵をサーバ1とクライアントサーバ2A, 2B, 2Cとの間で共有するために公開鍵暗号方式を用いている。従って、サーバ1、およびクライアントサーバ2A, 2B, 2Cは、自己の公開鍵58を第三者である認証機関へ登録し、その登録によって自己の公開鍵58を認証機関で認証してもらった電子証明書50を保持し、その電子証明書50によって自己が認証機関によ

って認証された正規のサーバであることを証明する。証明対象ユーザ55およびユーザユニーク識別子56は、ユーザの個人情報に該当する部分である。

【0054】従って、電子証明書50を受信したサーバ1等は、電子証明書50を認証機関が発行した公開鍵によって復号すれば、電子証明書50を送信した相手が正規のサーバであるか否かを判別できる。

【0055】図4～図6は、サーバ1とクライアントサーバ2A, 2B, 2Cとの間のSSL暗号通信におけるセッション確立のフローチャートである。まず、図4に示すフローチャートについて説明する。サーバ1とクライアントサーバ2A, 2B, 2Cとの間の通信が開始されると(ステップS100)、クライアントサーバ2A, 2B, 2CのHandshake Protocol部22は、”ClientHello”メッセージをRecord Protocol部21を介して送信する(ステップS102)。この”ClientHello”メッセージは、通信プロトコルのバージョン、セッションID、暗号化アルゴリズム等の候補を含む。CA Proxy 4の通信プロトコルキャプチャ部41は、クライアントサーバ2A, 2B, 2Cからの”ClientHello”メッセージを受信し、その受信した”ClientHello”メッセージをサーバ1へ送信する。サーバ1のHandshake Protocol部12は、Record Protocol部11を介して”ClientHello”メッセージを受信する(ステップS104)。そして、Handshake Protocol部12は、受信した”ClientHello”メッセージに含まれるプロトコルバージョン、セッションID、および暗号アルゴリズムの候補から1つのプロトコルバージョン、セッションID、および暗号アルゴリズムを選択し、その選択したプロトコルバージョン、セッションID、および暗号アルゴリズムを”ServerHello”メッセージに含めてクライアントサーバ2A, 2B, 2Cへ送信する(ステップS106)。

【0056】CA Proxy 4の通信プロトコルキャプチャ部41は、サーバ1からの”ServerHello”メッセージを受信し、その受信した”ServerHello”メッセージをクライアントサーバ2A, 2B, 2Cへ送信する。クライアントサーバ2A, 2B, 2CのHandshake Protocol部22は、”ServerHello”メッセージをRecord Protocol部21を介して受信し、”ServerHello”メッセージに基づいてサーバ1が選択したプロトコルバージョン、セッションID、および暗号アルゴリズムを確認する(ステップS108)。これによって、サーバ1とクライアントサーバ2A, 2B, 2Cとの間の暗号通信方式が決定される。

【0057】その後、サーバ1のHandshake



Protocol部12は、"ServerCertificate"メッセージをRecord Protocol部11を介して送信する(ステップS110)。なお、この"ServerCertificate"は、サーバ1の電子証明書であり、通信関連の国際標準機構であるITU(International Telecommunication Union)で標準化されたX.509v3による標準仕様に従って作成されている。以下に述べる"ClientCertificate"も同様である。CA Proxy4の通信プロトコルキャプチャ部41は、サーバ1からの"ServerCertificate"を受信し、その受信した"ServerCertificate"メッセージをCRLチェック部42へ出力する。CRLチェック部42は、"ServerCertificate"メッセージを受取り、サーバ1の電子証明書を証明書廃棄リスト(CRL)と照合し、サーバ1の電子証明書が証明書廃棄リスト(CRL)に含まれるか否かをチェックする。そして、CRLチェック部42は、照合結果を通信プロトコルキャプチャ部41へ出力する(ステップS112)。

【0058】サーバ1の電子証明書が証明書廃棄リスト(CRL)に含まれる場合、通信プロトコルキャプチャ部41は、サーバ1の電子証明書が無効であることを示す無効通知をサーバ1へ送信する(ステップS114)。そして、サーバ1のHandshake Protocol部12は、Record Protocol部11を介して無効通知を受信し(ステップS116)、サーバ1とクライアントサーバ2A、2B、2Cとの通信は終了する(ステップS154)。つまり、サーバ1は、正規のサーバではないと判断されたので、サーバ1とクライアントサーバ2A、2B、2Cとの通信は終了し、クライアントサーバ2A、2B、2Cの重要な情報が不正なサーバへ漏洩するのを防止できる。

【0059】ステップS112において、サーバ1の電子証明書が証明書廃棄リスト(CRL)に含まれていないと判断されたとき、CA Proxy4の通信プロトコルキャプチャ部41は、サーバ1から受信した電子証明書をクライアントサーバ2A、2B、2Cへ送信し(ステップS118)、クライアントサーバ2A、2B、2CのHandshake Protocol部22は、Record Protocol部21を介してサーバ1の電子証明書を受信する(ステップS120)。これによって、サーバ1は正規のサーバであることが認証されるとともに、クライアントサーバ2A、2B、2Cは、サーバ1の公開鍵を取得する。

【0060】そして、サーバ1のHandshake Protocol部12は、クライアントサーバ2A、2B、2Cに対して電子証明書の送信を要求するか否かを判定する(ステップS122)。電子証明書の送信を

要求しないとき、図5に示すステップS132へ移行する。また、Handshake Protocol部12は、電子証明書の提出を要求すると判定したとき、"Certificate Request"をRecord Protocol部11を介して送信し、クライアントサーバ2A、2B、2CのHandshake Protocol部22は、CA Proxy4およびRecord Protocol部21を介して"Certificate Request"を受信し、それに対して"No Certificate"メッセージをCA Proxy4へ送信する(ステップS124)。

【0061】次に、図5に示すフローチャートについて説明する。CA Proxy4の通信プロトコルキャプチャ部41は、クライアントサーバ2A、2B、2Cから"No Certificate"メッセージを受信し、その受信した"No Certificate"メッセージを認証部43および代理応答部44へ出力する。そして、認証部43は、"No Certificate"メッセージを受取ると、保持しているCA Proxy4の電子証明書を代理応答部44へ出力する。代理応答部44は、通信プロトコルキャプチャ部41からの"No Certificate"メッセージを受けると、サーバ1に対して代理応答するか否かを判定する(ステップS126)。代理応答部44が代理応答しないと判定したとき、ステップS154へ移行し、通信は終了する。代理応答部44は、代理応答すると判定すると、認証部43から入力されたCA Proxy4の電子証明書を通信プロトコルキャプチャ部41を介してサーバ1へ送信する(ステップS128)。

【0062】サーバ1のHandshake Protocol部12は、Record Protocol部11を介してCA Proxy4からの電子証明書を受信し、その受信した電子証明書に基づいてCA Proxy4が正規のサーバであることを認証するとともに、クライアントサーバ2A、2B、2Cとの暗号通信に用いる公開鍵を取得する(ステップS130)。この場合、Handshake Protocol部12は、形式的にはCA Proxy4を正規のサーバとして認証するが、CA Proxy4はクライアントサーバ2A、2B、2Cに代わって電子証明書をサーバ1へ送信しているので、Handshake Protocol部12は、実質的にはクライアントサーバ2A、2B、2Cを正規のサーバとして認証する。また、CA Proxy4は、クライアントサーバ2A、2B、2Cが電子証明書を保持しない場合でもクライアントサーバ2A、2B、2Cに代わって自己の電子証明書をサーバ1へ送信するので、認証機関によって認証された電子証明書を保持しないクライアントでも正規のサーバとしてサーバ1との暗号通信が可能になる。

【0063】ステップS122で”No”が選択された後、またはステップS130の後、クライアントサーバ2A、2B、2CのHandshake Protocol部22は、48バイトの乱数を発生させ、その発生させた乱数をRecord Protocol部21へ出力する。Record Protocol部21は、入力された乱数をサーバ1の公開鍵Paで暗号化し、その暗号化した乱数を”ClientKeyExchange”メッセージとしてサーバ1へ送信する(ステップS132)。Handshake Protocol部22が発生した乱数は、サーバ1とクライアントサーバ2A、2B、2Cとの間でデータを暗号通信する際の共通鍵を生成するためのものであり、Handshake Protocol部22は、発生した乱数を用いて共通鍵を生成する。

【0064】一方、サーバ1のRecord Protocol部11は、CA Proxy4を介して”ClientKeyExchange”メッセージを受信し、暗号化された乱数を秘密鍵Saで復号する(ステップS134)。そして、Record Protocol部11は、復号した乱数をHandshake Protocol部12へ出力する。Handshake Protocol部12は、入力された48バイトの乱数を用いて共通鍵を生成する。

【0065】その後、クライアントサーバ2A、2B、2CのChange Cipher Spec Protocol部23は、ステップS136までにサーバ1とクライアント2A、2B、2Cとの間で合意された暗号通信方式に同意し、生成した共通鍵を認めることを示す”ChangeCipherSpec”メッセージを生成してRecord Protocol部21へ出力する。Record Protocol部21は、共通鍵KPaによって”ChangeCipherSpec”メッセージを暗号化した{ChangeCipherSpec}KPaを生成してクライアントサーバ2A、2B、2Cへ送信する(ステップS136)。

【0066】サーバ1のRecord Protocol部11は、CA Proxy4を介して{ChangeCipherSpec}KPaを受信し、共通鍵KPaを用いて{ChangeCipherSpec}KPaを復号する。そして、Record Protocol部11は、復号した”ChangeCipherSpec”メッセージをChange Cipher Spec Protocol部13へ出力する。Change Cipher Spec Protocol部13は、”ChangeCipherSpec”メッセージを受けて、クライアントサーバ2A、2B、2Cが暗号通信方式や共通鍵に同意したことを検知する(ステップS138)。そして、クライアントサーバ2A、2B、2CのHandshake Protocol部22は、Handshak

e Protocolの終了を表す”Finished”メッセージを生成してRecord Protocol部21へ出力する。Record Protocol部21は、上記で決めた暗号化仕様に従って共通鍵KPaで”Finished”メッセージを暗号化し、{Finished}KPaをサーバ1へ出力する(ステップS140)。

【0067】サーバ1のRecord Protocol部11は、CA Proxy4を介して{Finished}KPaを受信し、その受信した{Finished}KPaを共通鍵KPaによって復号する。そして、Record Protocol部11は、復号した”Finished”メッセージをHandshake Protocol部12へ出力する。Handshake Protocol部12は、”Finished”メッセージを受信する(ステップS142)。

【0068】最後に、図6に示すフローチャートについて説明する。その後、Change Cipher Spec Protocol部13は、”ChangeCipherSpec”メッセージを生成してRecord Protocol部11へ出力する。Record Protocol部11は、共通鍵KPaによって”ChangeCipherSpec”メッセージを暗号化し、その暗号化した{ChangeCipherSpec}KPaをクライアントサーバ2A、2B、2Cへ送信する(ステップS144)。

【0069】クライアントサーバ2A、2B、2CのRecord Protocol部21は、CA Proxy4を介して{ChangeCipherSpec}KPaを受信し、共通鍵KPaによって{ChangeCipherSpec}KPaを復号する。そして、Record Protocol部21は、復号した”ChangeCipherSpec”メッセージをChange Cipher Spec Protocol部23へ出力する。Change Cipher Spec Protocol部23は、”ChangeCipherSpec”メッセージを受けてサーバ1が暗号通信方式や共通鍵に同意したことを検知する(ステップS146)。

【0070】その後、Handshake Protocol部12は、クライアントサーバ2A、2B、2Cへ送信するHandshake Protocolの終了を示す”Finished”メッセージを生成してRecord Protocol部11へ出力する。Record Protocol部11は、上記で決めた暗号化仕様に従って共通鍵KPaによって”Finished”メッセージを暗号化し、{Finished}KPaをクライアントサーバ2A、2B、2Cへ出力する(ステップS148)。クライアントサーバ2A、2B、2CのRecord Protocol部21は、

CA Proxy 4を介して {Finished} KPaを受信し、その受信した {Finished} KPaを共通鍵KPaによって復号する。そして、Record Protocol部21は、復号した"Finished"メッセージをHandshake Protocol部22へ出力する。Handshake Protocol部22は、"Finished"メッセージを受信する(ステップS150)。

【0071】ステップS150まででサーバ1とクライアントサーバ2A、2B、2CとのHandshake Protocol、即ち、セッションの確立が終了する。そして、サーバ1のApplication Data Protocol部15とアプリケーション部16、およびクライアントサーバ2A、2B、2CのApplication Data Protocol部25とアプリケーション部26による共通鍵KPaを用いた暗号通信が行なわれて(ステップS152)、サーバ1とクライアントサーバ2A、2B、2Cとの間の通信が終了する(ステップS154)。

【0072】図4から図6に示したフローチャートは、クライアントサーバ2A、2B、2Cが自己の電子証明書を保持しない場合のサーバ1とクライアントサーバ2A、2B、2Cとのセッション確立時のフローチャートである。上述したように、クライアントサーバ2A、2B、2Cが自己の電子証明書を保持しない場合でも、CA Proxy 4は、クライアントサーバ2A、2B、2Cが正規のサーバであることを示すCA Proxy 4の電子証明書をサーバ1へ代理応答し、サーバ1とクライアントサーバ2A、2B、2Cとの間で相互認証が行なわれる。

【0073】図7～図9は、サーバ1とクライアントサーバ2A、2B、2Cとのセッション確立時の別のフローチャートである。まず、図7に示すフローチャートについて説明する。サーバ1とクライアントサーバ2A、2B、2Cとの間の通信が開始されると(ステップS200)、クライアントサーバ2A、2B、2CのHandshake Protocol部22は、"ClientHello"メッセージをRecord Protocol部21を介して送信する(ステップS202)。CA Proxy 4の通信プロトコルキャプチャ部41は、クライアントサーバ2A、2B、2Cからの"ClientHello"メッセージを受信し、その受信した"ClientHello"メッセージをサーバ1へ送信する。サーバ1のHandshake Protocol部12は、Record Protocol部11を介して"ClientHello"メッセージを受信する(ステップS204)。そして、Handshake Protocol部12は、受信した"ClientHello"メッセージに含まれるプロトコルバージョン、セッションID、および暗号アルゴリズム

ズの候補から1つのプロトコルバージョン、セッションID、および暗号アルゴリズムを選択し、その選択したプロトコルバージョン、セッションID、および暗号アルゴリズムを"ServerHello"メッセージに含めてクライアントサーバ2A、2B、2Cへ送信する(ステップS206)。

【0074】CA Proxy 4の通信プロトコルキャプチャ部41は、サーバ1からの"ServerHello"メッセージを受信し、その受信した"ServerHello"メッセージをクライアントサーバ2A、2B、2Cへ送信する。クライアントサーバ2A、2B、2CのHandshake Protocol部22は、"ServerHello"メッセージをRecord Protocol部21を介して受信し、"ServerHello"メッセージに基づいてサーバ1が選択したプロトコルバージョン、セッションID、および暗号アルゴリズムを確認する(ステップS208)。これによって、サーバ1とクライアントサーバ2A、2B、2Cとの間の暗号通信方式が決定される。

【0075】その後、サーバ1のHandshake Protocol部12は、電子証明書を保持するか否かを判定し(ステップS210)、電子証明書を保持していないと判定したとき、ステップS224へ移行する。Handshake Protocol部12は、電子証明書を保持していると判定したとき、"ServerCertificate"メッセージをRecord Protocol部11を介して送信する(ステップS212)。CA Proxy 4の通信プロトコルキャプチャ部41は、サーバ1からの"ServerCertificate"を受信し、その受信した"ServerCertificate"メッセージをCRLチェック部42へ出力する。CRLチェック部42は、"ServerCertificate"メッセージを受取り、サーバ1の電子証明書を証明書廃棄リスト(CRL)と照合し、サーバ1の電子証明書が証明書廃棄リスト(CRL)に含まれるか否かをチェックする。そして、CRLチェック部42は、照合結果を通信プロトコルキャプチャ部41へ出力する(ステップS214)。

【0076】サーバ1の電子証明書が証明書廃棄リスト(CRL)に含まれる場合、通信プロトコルキャプチャ部41は、サーバ1の電子証明書が無効であることを示す無効通知をサーバ1へ送信する(ステップS216)。そして、サーバ1のHandshake Protocol部12は、Record Protocol部11を介して無効通知を受信し(ステップS218)、サーバ1とクライアントサーバ2A、2B、2Cとの通信は終了する(ステップS276)。つまり、サーバ1は、正規のサーバではないと判断されたので、サーバ1とクライアントサーバ2A、2B、2Cとの通信

は終了し、クライアントサーバ2A、2B、2Cの重要な情報が不正なサーバへ漏洩するのを防止できる。

【0077】ステップS214において、サーバ1の電子証明書が証明書廃棄リスト(CRL)に含まれていないと判断されたとき、CA Proxy4の通信プロトコルキャプチャ部41は、サーバ1から受信した電子証明書をクライアントサーバ2A、2B、2Cへ送信し(ステップS220)、クライアントサーバ2A、2B、2CのHandshake Protocol部22は、Record Protocol部21を介してサーバ1の電子証明書を受信する(ステップS222)。これによって、サーバ1は正規のサーバであることが認証されるとともに、クライアントサーバ2A、2B、2Cは、サーバ1の公開鍵を取得する。

【0078】一方、ステップS210においては、Handshake Protocol部12が電子証明書を保持していないと判定したとき、Handshake Protocol部12は、“ServerKeyExchange”メッセージを生成してRecord Protocol部11へ出力する。そして、Record Protocol部11は、“ServerKeyExchange”メッセージをクライアントサーバ2A、2B、2Cへ送信する(ステップS224)。この“ServerKeyExchange”メッセージは、RSA公開鍵またはDiffie&Hellman公開情報から成る。サーバ1が電子証明書を保持しない場合に、RSA公開鍵またはDiffie&Hellman公開情報をクライアントサーバ2A、2B、2Cへ送信することによってサーバ1が正規のサーバであることを電子証明書に代えて証明するものである。

【0079】ステップS222またはステップS224の後、サーバ1のHandshake Protocol部12は、クライアントサーバ2A、2B、2Cに対して電子証明書の送信を要求するか否かを判定する(ステップS226)。電子証明書の送信を要求しないとき、図9に示すステップS254へ移行する。また、Handshake Protocol部12は、電子証明書の提出を要求すると判定したとき、“Certificate Request”をRecord Protocol部11を介して送信し、クライアントサーバ2A、2B、2CのHandshake Protocol部22は、CA Proxy4およびRecord Protocol部21を介して“Certificate Request”を受信する。

【0080】次に、図8に示すフローチャートについて説明する。クライアントサーバ2A、2B、2CのHandshake Protocol部22は、電子証明書を保持しているか否かを判定する(ステップS228)。そして、Handshake Protocol部22は、電子証明書を保持していると判定したと

き、“ClientCertificate”メッセージをRecord Protocol部21を介してCA Proxy4へ送信する(ステップS230)。CA Proxy4の通信プロトコルキャプチャ部41は、クライアントサーバ2A、2B、2Cから“ClientCertificate”メッセージを受信し、その受信した“ClientCertificate”メッセージを認証部43および代理応答部44へ出力する。そして、認証部43は、“ClientCertificate”メッセージを受取ると、そのメッセージに含まれている電子証明書に基づいてクライアントサーバ2A、2B、2Cを正規のサーバとして認証し、保持しているCA Proxy4の電子証明書を代理応答部44へ出力する(ステップS232)。

【0081】ステップS228において、Handshake Protocol部22が電子証明書を保持していないと判定したとき、Handshake Protocol部22は、“NoCertificate”メッセージをCA Proxy4へ送信する(ステップS234)。そして、CA Proxy4の通信プロトコルキャプチャ部41は、クライアントサーバ2A、2B、2Cから“NoCertificate”メッセージを受信し、その受信した“NoCertificate”メッセージを認証部43および代理応答部44へ出力する。認証部43は、“NoCertificate”メッセージを受取ると、保持しているCA Proxy4の電子証明書を代理応答部44へ出力する(ステップS236)。

【0082】ステップS232またはステップS236の後、代理応答部44は、通信プロトコルキャプチャ部41からの“ClientCertificate”メッセージまたは“NoCertificate”メッセージを受けると、サーバ1に対して代理応答するか否かを判定する(ステップS238)。代理応答部44が代理応答しないと判定したとき、ステップS276へ移行し、通信は終了する。代理応答部44は、代理応答すると判定すると、認証部43から入力されたCA Proxy4の電子証明書を通信プロトコルキャプチャ部41を介してサーバ1へ送信する(ステップS240)。

【0083】サーバ1のHandshake Protocol部12は、Record Protocol部11を介してCA Proxy4からの電子証明書を受信し、その受信した電子証明書に基づいてCA Proxy4が正規のサーバであることを認証するとともに、クライアントサーバ2A、2B、2Cとの暗号通信に用いる公開鍵を取得する(ステップS242)。この場合、Handshake Protocol部12は、上述したように、クライアントサーバ2A、2B、2Cを正規のサーバとして認証する。

【0084】その後、クライアントサーバ2A、2B、2CのHandshake Protocol部22は、48バイトの乱数を発生させ、その発生させた乱数をRecord Protocol部21へ出力する。Record Protocol部21は、入力された乱数を公開鍵Paで暗号化し、その暗号化した乱数を”ClientKeyExchange”メッセージとしてサーバ1へ送信する（ステップS244）。また、Handshake Protocol部22は、発生した乱数を用いてサーバ1とクライアントサーバ2A、2B、2Cとの間でデータを暗号通信する際の共通鍵を生成する。

【0085】一方、サーバ1のRecord Protocol部11は、CA Proxy4を介して”ClientKeyExchange”メッセージを受信し、暗号化された乱数を秘密鍵Saで復号する（ステップS246）。そして、Record Protocol部11は、復号した乱数をHandshake Protocol部12へ出力する。Handshake Protocol部12は、入力された48バイトの乱数を用いて共通鍵を生成する。

【0086】クライアントサーバ2A、2B、2CのHandshake Protocol部22は、再度、電子証明書を保持するか否かを判定する（ステップS248）。電子証明書が保持されているときステップS250へ移行し、保持されていないときステップS258へ移行する。

【0087】最後に、図9に示すフローチャートについて説明する。ステップS248において、電子証明書が保持されていると判定されたとき、Handshake Protocol部22は、サーバ1がクライアントサーバ2A、2B、2Cの電子証明書が正しいことを確認するために、ステップS248までに取得されたメッセージのダイジェストをRecord Protocol部21へ出力する。Record Protocol部21は、入力されたメッセージのダイジェストを秘密鍵Saで暗号化し、”CertificateVerify”としてサーバ1へ送信する（ステップS250）。サーバ1のRecord Protocol部11は、”CertificateVerify”をCA Proxy4を介して受信し、暗号化されたメッセージのダイジェストを公開鍵で復号する。そして、Record Protocol部11は、復号したメッセージのダイジェストをHandshake Protocol部12へ出力し、Handshake Protocol部12は、入力されたメッセージのダイジェストに基づいてクライアントサーバ2A、2B、2Cの電子証明書が正しいを確認する（ステップS252）。

【0088】一方、ステップS226（図7参照）において、サーバ1がクライアントサーバ2A、2B、2C

の電子証明書の送信を要求しないと判定としたとき、クライアントサーバ2A、2B、2CのHandshake Protocol部22は、48バイトの乱数を発生させ、その発生させた乱数をRecord Protocol部21へ出力する。Record Protocol部21は、入力された乱数をサーバ1の公開鍵Paで暗号化し、その暗号化した乱数を”ClientKeyExchange”メッセージとしてサーバ1へ送信する（ステップS254）。また、Handshake Protocol部22は、発生した乱数を用いてサーバ1とクライアントサーバ2A、2B、2Cとの間でデータを暗号通信する際の共通鍵を生成する。

【0089】一方、サーバ1のRecord Protocol部11は、CA Proxy4を介して”ClientKeyExchange”メッセージを受信し（、暗号化された乱数を秘密鍵Saで復号する（ステップS256）。そして、Record Protocol部11は、復号した乱数をHandshake Protocol部12へ出力する。Handshake Protocol部12は、入力された48バイトの乱数を用いて共通鍵を生成する。

【0090】ステップS248においてクライアントサーバ2A、2B、2Cで電子証明書が保持されていないと判定されたとき、またはステップS252、S256の後、クライアントサーバ2A、2B、2CのChange Cipher Spec Protocol部23は、ステップS256までにサーバ1とクライアント2A、2B、2Cとの間で合意された暗号通信方式に同意し、生成した共通鍵を認めることを示す”ChangeCipherSpec”メッセージを生成してRecord Protocol部21へ出力する。Record Protocol部21は、共通鍵KPaによって”ChangeCipherSpec”メッセージを暗号化した{ChangeCipherSpec}KPaを生成してクライアントサーバ2A、2B、2Cへ送信する（ステップS258）。

【0091】それ以後のステップS260～S276は、図5および図6のステップS138～154と同じである。

【0092】図7～図9に示すフローチャートは、サーバ1が電子証明書を保持しない場合に、RSA公開鍵またはDiffie&Hellman公開情報をクライアントサーバ2A、2B、2Cへ送信してサーバ1の正当性を認証してもらう点が、図4～図6に示したフローチャートと特に異なる点である。

【0093】上記においては、サーバとクライアントサーバとの間の暗号通信方式をSSL暗号通信として説明したが、本発明は、これに限られるものではなく、公開鍵暗号方式であれば、どのような暗号方式を用いたものであっても良い。

【0094】この発明の実施の形態によれば、暗号通信システムは、サーバに対してクライアントサーバが正規のサーバであることを代理応答するCA Proxyを備えるので、クライアントの個人情報がサーバに公開されることなく、サーバとクライアントサーバとの間で相互に認証し、暗号通信を行なうことができる。

【0095】今回開示された実施の形態はすべての点で例示であって制限的なものではないと考えられるべきである。本発明の範囲は上記した説明ではなくて特許請求の範囲によって示され、特許請求の範囲と均等の意味および範囲内でのすべての変更が含まれることが意図される。

#### 【0096】

【発明の効果】この発明による暗号通信システムは、サーバに対してクライアントサーバが正規のサーバであることを自己の電子証明書を用いて代理応答するCA Proxyを備えるので、クライアントの個人情報がサーバに公開されることなく、サーバとクライアントサーバとの間で相互に認証し、暗号通信を行なうことができる。

#### 【図面の簡単な説明】

【図1】 本発明の実施の形態による暗号通信システムの概略ブロック図である。

【図2】 図1に示すサーバ、クライアントサーバ、およびCA Proxyの機能ブロック図である。。

【図3】 電子証明書の構成を示す概略ブロック図である。

【図4】 サーバとクライアントサーバとの間のセッション確立時の第1のフローチャートである。

【図5】 サーバとクライアントサーバとの間のセッション確立時の第2のフローチャートである。

【図6】 サーバとクライアントサーバとの間のセッション確立時の第3のフローチャートである。

【図7】 サーバとクライアントサーバとの間の他の方

法によるセッション確立時の第1のフローチャートである。

【図8】 サーバとクライアントサーバとの間の他の方法によるセッション確立時の第2のフローチャートである。

【図9】 サーバとクライアントサーバとの間の他の方法によるセッション確立時の第3のフローチャートである。

【図10】 OSI参照モデルを用いたコンピュータ間の通信するための概略図である。

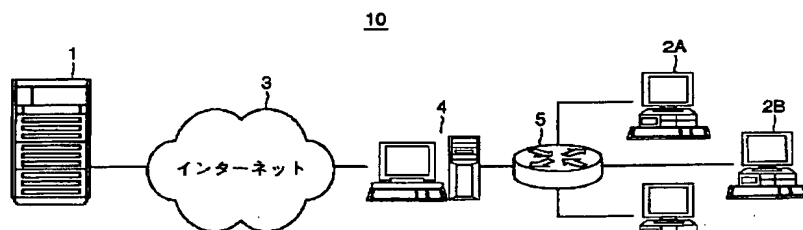
【図11】 OSI参照モデルの各層の機能を説明するための図表である。

【図12】 従来のSSL代理応答システムの概略ブロック図である。

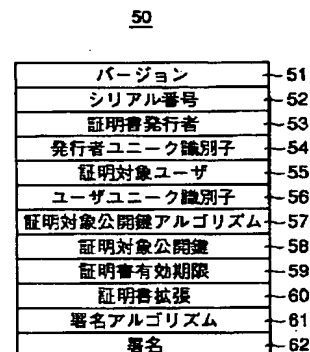
#### 【符号の説明】

1, 240 サーバ、2A, 2B, 2C クライアントサーバ、3 インターネット網、4 CA Proxy、5 結合器、10 暗号通信システム、11, 21 Record Protocol部、12, 22 Handshake Protocol部、13, 23 Change Cipher Spec Protocol部、14, 24 Alert Protocol部、15, 25 Application Data Protocol部、16, 26 アプリケーション部、41 通信プロトコルキャプチャ部、42 CRLチェック部、43 認証部、44 代理応答部、50 電子証明書、51 バージョン、52 シリアル番号、53 証明書発行者、54 発行者ユニーク識別子、55 証明対象ユーザ、56 ユーザユニーク識別子、57 証明対象公開鍵アルゴリズム、58 証明対象公開鍵、59 証明書有効期限、60 証明書拡張、61 署名アルゴリズム、62 署名、200 SSL代理応答システム、210 インターネット、220 ファイアウォール、230 代理サーバ。

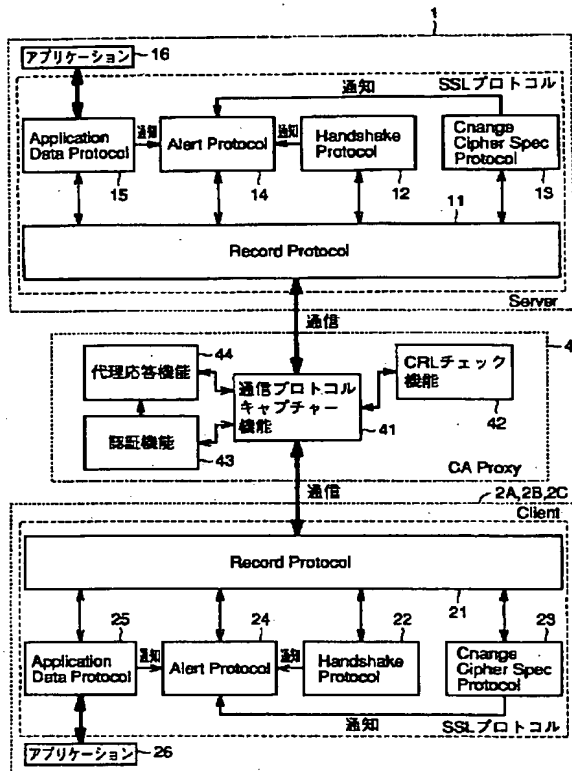
【図1】



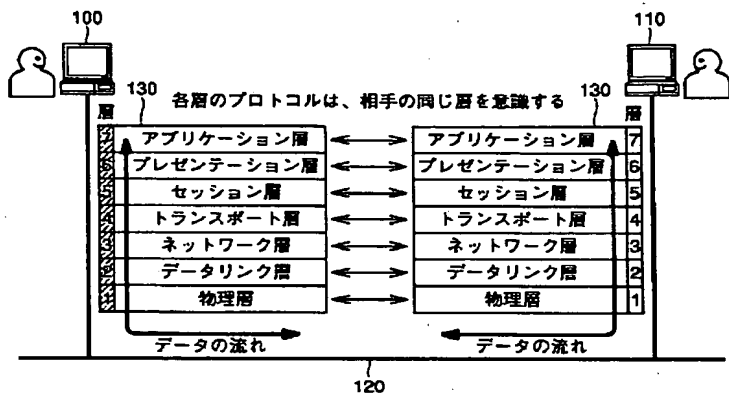
【図3】



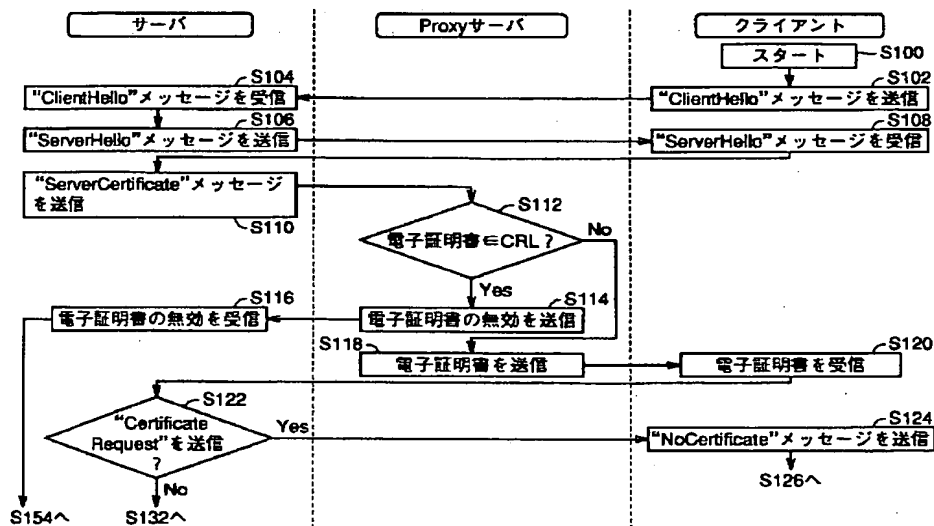
【図2】



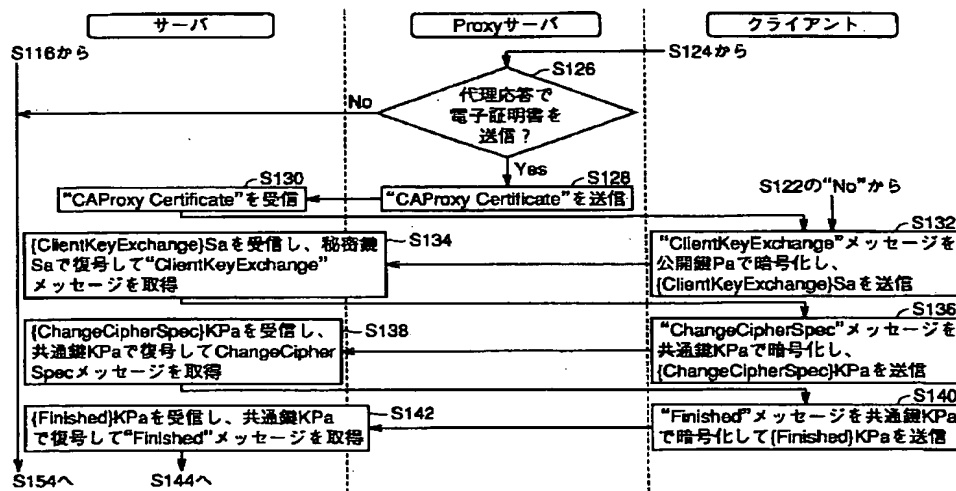
【図10】



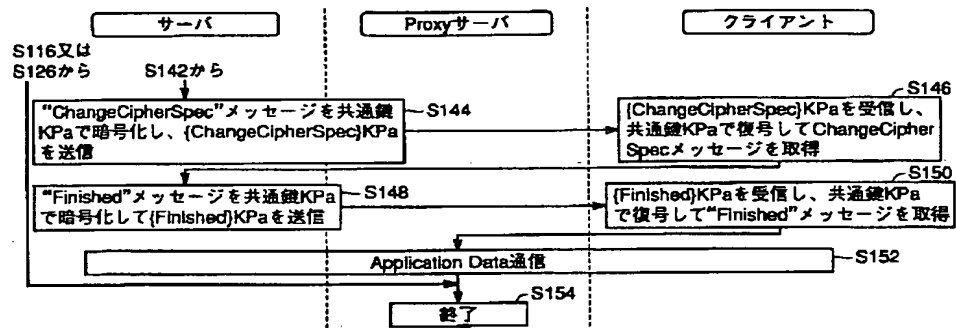
【図4】



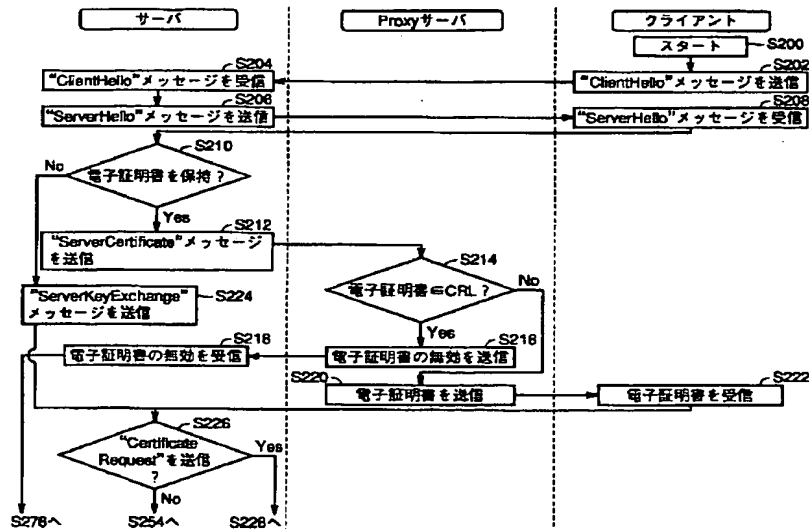
【図5】



【図6】



【図7】





```

sequenceDiagram
    participant S as サーバ
    participant P as Proxyサーバ
    participant C as クライアント
    Note over C: S228から
    C->>P: S228
    P->>C: S228
    C-->>P: Yes
    C-->>P: No
    P->>S: S232 "ClientCertificate"メッセージを受信
    P->>S: S236 "NoCertificate"メッセージを受信
    S->>P: S238
    S-->>P: No
    S-->>P: Yes
    P->>S: S240 "CAProxy Certificate"を送信
    P->>C: S242 "CAProxy Certificate"を受信
    P->>C: S244 "ClientKeyExchange"メッセージを公開鍵Paで暗号化し、(ClientKeyExchange)Saを送信
    C->>P: S246 (ClientKeyExchange)Saを受信し、秘密鍵Saで復号して"ClientKeyExchange"メッセージを取得
    P->>S: S248
    S->>P: S250
    S->>C: S276へ
    
```

```

sequenceDiagram
    participant S as サーバ
    participant PS as Proxyサーバ
    participant C as クライアント

    S->>PS: S218又はS238の"No"から
    PS->>C: S248の"Yes"から
    PS->>C: S248の"No"から
    PS->>C: S226の"No"から

    S->>PS: S252 "CertificateVerify"メッセージを受信
    PS->>C: S250 "CertificateVerify"メッセージを送信

    PS->>S: S256 {ClientKeyExchange}Saを受信し、秘密鍵Saで復号して{ClientKeyExchange}メッセージを取得
    PS->>C: S254 "ClientKeyExchange"メッセージを公開鍵Paで暗号化し、{ClientKeyExchange}Saを送信

    S->>PS: S260 {ChangeCipherSpec}KPaを受信し、共通鍵KPaで復号してChangeCipherSpecメッセージを取得
    PS->>C: S258 "ChangeCipherSpec"メッセージを共通鍵KPaで暗号化し、{ChangeCipherSpec}KPaを送信

    S->>PS: S264 {Finished}KPaを受信し、公開鍵KPaで復号して{Finished}メッセージを取得
    PS->>C: S262 {Finished}メッセージを共通鍵KPaで暗号化して{Finished}KPaを送信

    PS->>S: S266 {ChangeCipherSpec}メッセージを共通鍵KPaで暗号化し、{ChangeCipherSpec}Pbを送信
    PS->>C: S268 {ChangeCipherSpec}K Pbを受信し、共通鍵KPaで復号して{ChangeCipherSpec}メッセージを取得

    S->>PS: S270 {Finished}メッセージを共通鍵KPaで暗号化して{Finished}KPaを送信
    PS->>C: S272 {Finished}KPaを受信し、共通鍵KPaで復号して{Finished}メッセージを取得

    PS->>S: S274 Application Data通信
    PS->>S: S276 終了
  
```

The diagram illustrates the TLS handshake process between a Server (サーバ), Proxy Server (Proxyサーバ), and Client (クライアント). The process is divided into two main phases: the handshake phase and the application data phase.

**Handshake Phase:**

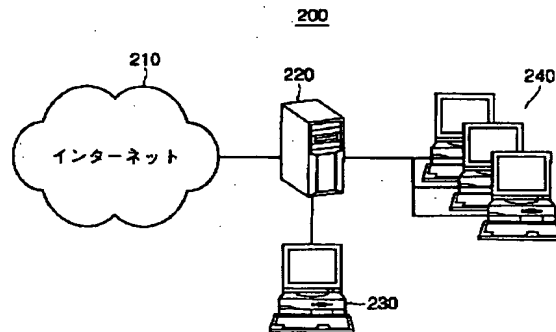
- Initial State:** The Proxy Server receives a "No" response from the Server (S218 or S238) and the Client (S248 "Yes" or "No", S226 "No").
- Certificate Verification:** The Server sends a "CertificateVerify" message to the Proxy Server (S252). The Proxy Server then sends a "CertificateVerify" message to the Client (S250).
- Client Key Exchange:** The Proxy Server receives a {ClientKeyExchange}Sa message from the Server (S256), decrypts it with the secret key Sa, and extracts the {ClientKeyExchange} message. The Proxy Server then sends a "ClientKeyExchange" message to the Client (S254), which is encrypted with the public key Pa and contains the {ClientKeyExchange}Sa.
- Change Cipher Spec:** The Server sends a {ChangeCipherSpec}KPa message to the Proxy Server (S260), which is decrypted with the common key KPa to extract the {ChangeCipherSpec} message. The Proxy Server then sends a "ChangeCipherSpec" message to the Client (S258), which is encrypted with the common key KPa and contains the {ChangeCipherSpec}KPa.
- Finished:** The Server sends a {Finished}KPa message to the Proxy Server (S264), which is decrypted with the public key KPa to extract the {Finished} message. The Proxy Server then sends a {Finished} message to the Client (S262), which is encrypted with the common key KPa and contains the {Finished}KPa.
- Server Key Exchange:** The Proxy Server sends a {ChangeCipherSpec} message to the Server (S266), which is encrypted with the common key KPa and contains the {ChangeCipherSpec}Pb. The Proxy Server then receives a {ChangeCipherSpec}K Pb message from the Client (S268), which is decrypted with the common key KPa to extract the {ChangeCipherSpec} message.
- Final Finished:** The Server sends a {Finished} message to the Proxy Server (S270), which is encrypted with the common key KPa and contains the {Finished}KPa. The Proxy Server then receives a {Finished}KPa message from the Client (S272), which is decrypted with the common key KPa to extract the {Finished} message.

**Application Data Phase:**

- The Proxy Server sends "Application Data通信" (S274) to the Server.
- The Proxy Server sends "終了" (S276) to the Server, indicating the end of the process.

OSI参照モデル上の分類			機能概要
	層	層名称	
上位層	第7層	アプリケーション層 (application layer)	ファイル転送やメッセージ通信(E-mail)など、ユーザが実行する多くのサービス間プロトコルを制御
	第6層	プレゼンテーション層 (presentation layer)	文字コードや画像データの表現形式を制御し、プロセス間におけるデータ形式などを確認する
	第5層	セッション層 (session layer)	アプリケーションプロセス間の情報の流れなど、通信モードの管理や情報転送に関する通信制御
下位層	第4層	トランスポート層 (transport layer)	通信情報の質を高めるための通信制御などを行う。データに抜けがあった場合、相手に通知する
	第3層	ネットワーク層 (network layer)	複数のネットワークにまたがったコンピュータ間のデータ転送やデータの中間機能など
	第2層	データリンク層 (data link layer)	ノード間で信頼性の高いデータ伝送を保証。中継局間のデータ伝送を確実に行う
	第1層	物理層 (physical layer)	データを電気信号に変換し、実際の伝送を行う

【図12】



## 【手続補正書】

【提出日】平成13年2月27日（2001. 2. 27）

【補正対象項目名】図5

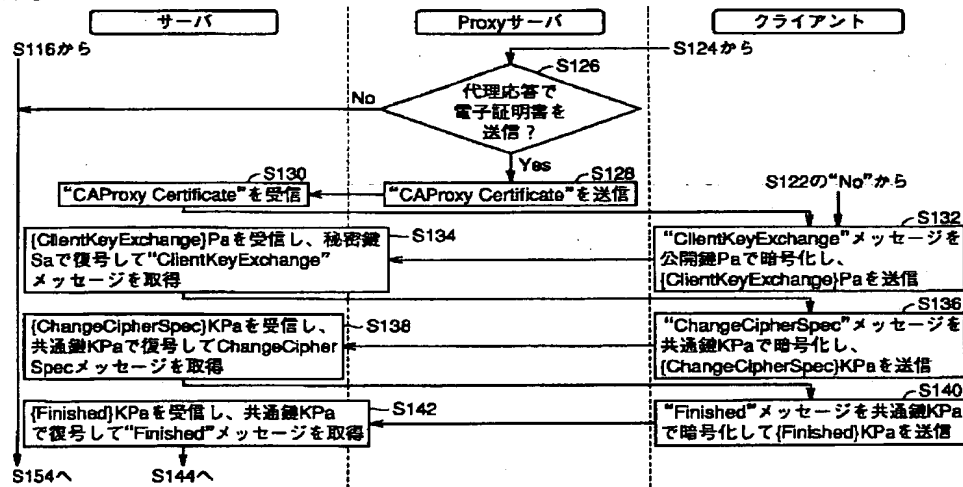
【補正方法】変更

## 【手続補正1】

【補正内容】

【補正対象書類名】図面

【図5】



## 【手続補正2】

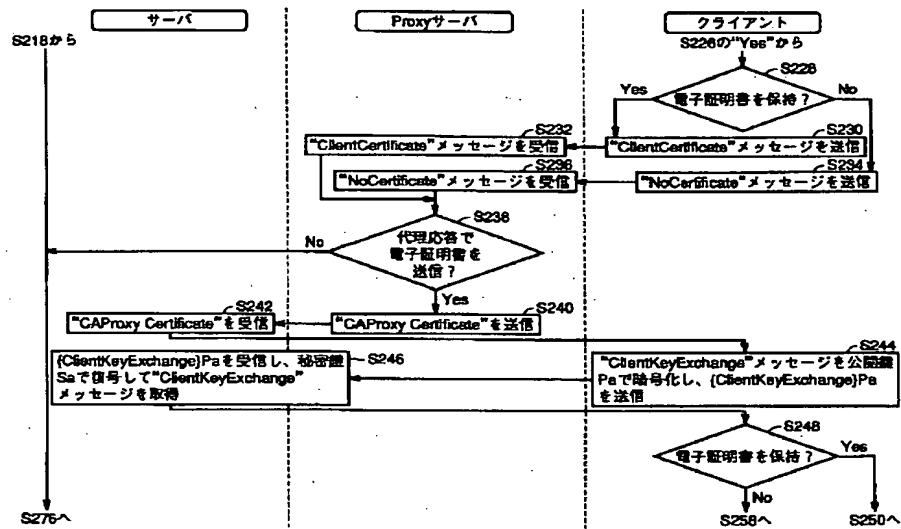
【補正方法】変更

【補正対象書類名】図面

【補正内容】

【補正対象項目名】図8

【図8】



## 【手続補正3】

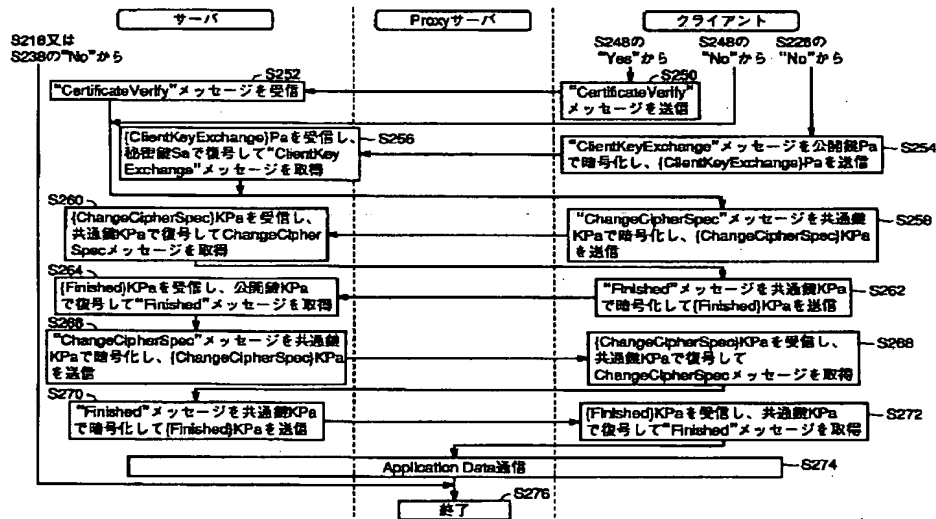
【補正対象書類名】図面

【補正対象項目名】図9

## 【補正方法】変更

【補正内容】

【図9】



フロントページの続き

(72) 発明者 山崎 達也  
 京都府相楽郡精華町光台二丁目2番地2  
 株式会社エイ・ティ・アール環境適応通信  
 研究所内

Fターム(参考) 5J104 AA01 AA07 EA05 JA21 KA02  
 MA01 NA02 PA07